Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 15 of 15 returned.

☐ 1. Document ID: US 6738870 B2

Using default format because multiple data bases are involved.

L1: Entry 1 of 15 File: USPT May 18, 2004

US-PAT-NO: 6738870
DOCUMENT-IDENTIFIER: US 6738870 B2

TITLE: High speed remote storage controller

DATE-ISSUED: May 18, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Van Huben; Gary A.	Poughkeepsie	NY		
Blake; Michael A.	Wappingers Falls	NY		
Mak; Pak-Kin	Poughkeepsie	NY		

US-CL-CURRENT: 711/150; 711/124, 711/130, 711/147, 711/148

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

☐ 2. Document ID: US 6714957 B1

L1: Entry 2 of 15 File: USPT Mar 30, 2004

DOCUMENT-IDENTIFIER: US 6714957 B1
TITLE: System and method for efficient processing of denormal results as hardware exceptions

Drawing Description Text (4):
FIG. 2 illustrates in more detail the exemplary integrated processor in FIG. 1, including the CPU, graphics controller, memory controller, and L2 unified cache, according to one embodiment of the present invention;

Detailed Description Text (7):
FIG. 2 illustrates in more detail selected portions of exemplary integrated processor 100, including CPU 105, graphics controller 110, memory controller 115, and L2 unified cache 120 (e.g., 256 KB in size). CPU 105 includes an execution pipeline with instruction decode/dispatch logic 200 and functional units 250.

3. Document ID: US 6629231 B1

L1: Entry 3 of 15 File: USPT Sep 30, 2003

DOCUMENT-IDENTIFIER: US 6629231 B1
TITLE: System and method for efficient register file conversion of denormal numbers between scalar and SIMD formats

Drawing Description Text (4):
FIG. 2 illustrates in more detail the exemplary integrated processor in FIG. 1, including the CPU, graphics controller, memory controller, and L2 unified cache according to one embodiment of the present invention;

Detailed Description Text (7):
FIG. 2 illustrates in more detail exemplary integrated processor 100, including CPU 105, which is integrated with graphics controller 110, memory controller 115, and L2 unified cache 120 (e.g., 256 KB in size). CPU 105 includes an execution pipeline with instruction decode/dispatch logic 200 and functional units 250.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. De

4. Document ID: US 6598064 B1

L1: Entry 4 of 15 File: USPT Jul 22, 2003

DOCUMENT-IDENTIFIER: US 6598064 B1
TITLE: Split multiplier array and method of operation

Drawing Description Text (4):
FIG. 2 illustrates in more detail the exemplary integrated processor in FIG. 1, including the CPU, graphics controller, memory controller, and L2 unified cache according to one embodiment of the present invention;

Detailed Description Text (8):
FIG. 2 illustrates in more detail exemplary integrated processor 100, including CPU 105, which is integrated with graphics controller 110, memory controller 115, and L2 unified cache 120 (e.g., 256 KB in size). CPU 105 includes an execution pipeline with instruction decode/dispatch logic 200 and functional units 250.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. De

5. Document ID: US 6581155 B1

L1: Entry 5 of 15 File: USPT Jun 17, 2003

DOCUMENT-IDENTIFIER: US 6581155 B1
TITLE: Pipelined, superscalar floating point unit having out-of-order execution capability and processor employing the same

Detailed Description Text (7):
Turning now to FIG. 2, illustrated in more detail is the exemplary integrated microprocessor 100, including CPU 110 integrated with graphics controller 120, memory controller 130, and L2 unified cache 140 (256 KB). CPU 110 includes an execution pipeline with instruction decode/dispatch logic 200 and functional units 250.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

☐ 6. Document ID: US 6535946 B1

L1: Entry 6 of 15 File: USPT Mar 18, 2003

DOCUMENT-IDENTIFIER: US 6535946 B1
TITLE: Low-latency circuit for synchronizing data transfers between clock domains derived from a common clock

Detailed Description Text (8):
FIG. 2 illustrates in more detail the exemplary integrated processor 100, including CPU 110, which is integrated with graphics controller 120, memory controller 130, and L2 unified cache 140 (e.g., 256 KB in size). CPU 110 includes an execution pipeline with instruction decode/dispatch logic 200 and functional units 250.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

☐ 7. Document ID: US 6415367 B1

L1: Entry 7 of 15 File: USPT Jul 2, 2002

DOCUMENT-IDENTIFIER: US 6415367 B1
TITLE: Apparatus for reducing asynchronous service latency in a time slot-based memory arbitration scheme

Brief Summary Text (5):
Computer systems that employ a CPU often utilize a memory controller and a graphics controller. The memory controller controls access by the CPU and other agents to a system memory. The graphics controller controls the display of data provided by the CPU onto a display screen, such as a cathode ray tube (CRT), using a frame buffer. Both the system memory and the frame buffer are typically implemented using arrays of Dynamic Random Access Memory (DRAM). In some computer systems, the frame buffer and the system memory are unified into a single shared memory, known as a Unified Memory Architecture (UMA).

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw. De

---

8. Document ID: US 6412063 B1

L1: Entry 8 of 15 File: USPT Jun 25, 2002

DOCUMENT-IDENTIFIER: US 6412063 B1
TITLE: Multiple-operand instruction in a two operand pipeline and processor employing the same

Detailed Description Text (7):
FIG. 2 illustrates in more detail the exemplary integrated microprocessor 100, including CPU 110 integrated with graphics controller 120, memory controller 130, and L2 unified cache 140 (256 KB). CPU 110 includes an execution pipeline with instruction decode/dispatch logic 200 and functional units 250.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw. De

---

9. Document ID: US 6412049 B1

L1: Entry 9 of 15 File: USPT Jun 25, 2002

DOCUMENT-IDENTIFIER: US 6412049 B1
TITLE: Method for minimizing CPU memory latency while transferring streaming data

Brief Summary Text (5):
Computer systems that employ a CPU often utilize a memory controller and a graphics controller. The memory controller controls access by the CPU and other agents to a system memory. The graphics controller controls the display of data provided by the CPU onto a display screen, such as a cathode ray tube (CRT), using a frame buffer. Both the system memory and the frame buffer are typically implemented using arrays of Dynamic Random Access Memory (DRAM). In some computer systems, the frame buffer and the system memory are unified into a single shared memory, known as a Unified Memory Architecture (UMA).

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw. De

---

10. Document ID: US 6363461 B1

L1: Entry 10 of 15 File: USPT Mar 26, 2002

DOCUMENT-IDENTIFIER: US 6363461 B1
TITLE: Apparatus for memory resource arbitration based on dedicated time slot allocation

Brief Summary Text (5):
Computer systems that employ a CPU often utilize a memory controller and a graphics controller. The memory controller controls access by the CPU and other agents to a system memory. The graphics controller controls the display of data provided by the CPU onto a display screen, such as a cathode ray tube (CRT), using a frame buffer. Both the system memory and the frame buffer are typically implemented using arrays of Dynamic Random Access Memory (DRAM). In some computer systems, the frame buffer and the system memory are unified into a single shared memory, known as a Unified Memory Architecture (UMA).

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. De

11. Document ID: US 6275926 B1

L1: Entry 11 of 15 File: USPT Aug 14, 2001

DOCUMENT-IDENTIFIER: US 6275926 B1
TITLE: System and method for writing back multiple results over a single-result bus and processor employing the same

Detailed Description Text (7):
FIG. 2 illustrates in more detail the exemplary integrated microprocessor 100, including CPU 110 integrated with graphics controller 120, memory controller 130, and L2 unified cache 140 (256 KB). CPU 110 includes an execution pipeline with instruction decode/dispatch logic 200 and functional units 250.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. De

12. Document ID: US 6222564 B1

L1: Entry 12 of 15 File: USPT Apr 24, 2001

DOCUMENT-IDENTIFIER: US 6222564 B1
**** See image for Certificate of Correction ****
TITLE: Method and apparatus for managing access to a computer system memory shared by a graphics controller and a memory controller

Abstract Text (1):
The shared computer system memory is partitioned between system memory and frame buffer memory. The frame buffer is configured as the top-most portion of the shared memory. A virtual memory manager controls access to the system portion of the shared memory. A virtual frame buffer device controls access to the frame buffer portion of the shared memory. While a frame buffer is defined, graphics operations, such as commands data, generated by a host CPU are routed to the memory manager by the virtual frame buffer device driver. Other graphics operations, such as those provided by peripheral devices interconnected via a PCI bus, are executed by the graphics controller. If an input buffer of the memory controller is full, graphics operations issued by the host CPU are rerouted onto the PCI bus for execution by

the graphics controller. If no frame buffer is defined, then all graphics operations generated by the host CPU as well as all other PCI bus masters are transmitted over the PCI bus and are executed by the graphics controller. While a frame buffer is defined, memory access protocols and cache coherency protocols, described herein, are employed to prevent corruption of the data within the frame buffer portion of the shared memory. In one embodiment, the memory controller and the graphics controller both access the shared memory through a single interface bus. In an alternative embodiment also described herein, the shared memory is physically divided into two portions. The memory controller is interconnected by a first interface bus to both portions of the shared memory. The graphics controller is connected to only the second portion of the memory which is configured to include a frame buffer. As such, the memory controller may access system memory contained within the first portion of the shared memory while the graphics controller is accessing the frame buffer. In another alternative embodiment described herein, no graphics controller is provided. Rather, all graphics operations are performed directly by the host CPU. A frame buffer refresh unit takes the place of the graphics controller.

Brief Summary Text (10):
Hence, in such a system, both the memory controller and the graphics controller can access the same array of physical memory through a single interface bus. An arbitration mechanism must be employed to prevent conflicts between the memory controller and the graphics controller. An appropriate arbitration mechanism is set forth in the above-referenced co-pending patent application. A management system should also be employed to control access to the shared memory by software running on the microprocessor including operating system software and applications software. For example, operating system software must be prevented from corrupting the frame buffer portion of the shared memory while that portion of the shared memory is being used as a frame buffer by the graphics controller. Such corruption could occur if, for example, operating system software swaps data into or out of the frame buffer portion of the shared memory. On the other hand, to gain the benefit of having additional system memory when a frame buffer is not required, application software should be able to access all portions of the shared memory.

Brief Summary Text (14):
Another problem with implementing a shared computer system memory accessible through only a single interface bus is that access to the memory by the memory controller must be frequently blocked to allow the graphics controller to perform frame buffer refresh operations. In a typical proposed implementation, such frame buffer refresh operations must be performed fairly frequently. Accordingly, considerable bus bandwidth must be devoted to handling frame buffer refresh operations. Accordingly, it would be desirable to provide an improved method and apparatus for partitioning the system memory and the frame buffer memory within the shared memory to minimize the impact on total bus bandwidth by frame buffer refresh operations. Other aspects of the invention are drawn to such an improved partitioning.

Brief Summary Text (16):
A method and apparatus for managing access to a shared computer system memory accessible by both a memory controller and a graphics controller through a single interface bus is provided. In accordance with one aspect of the invention, the computer system is configured to selectively define a portion of the shared memory as a frame buffer. Graphics commands and data provided a microprocessor or other client device are routed through the memory controller if a frame buffer is created. Otherwise, graphics commands and data are routed through the graphics controller.

Brief Summary Text (17):
In one embodiment, the shared memory is partitioned between a system memory portion and a frame buffer memory portion with the frame buffer defined within a

predetermined top-most portion of the physical memory. For example, in an eight megabyte memory system, the frame buffer may be defined as residing within the top one megabyte of the memory. At system start-up, a BIOS transmits a signal to an operating system of the microprocessor identifying the top of system memory as being the bottom of the frame buffer portion of memory. Accordingly, thereafter, operating system memory commands access only the system memory portion of the shared memory. Access to the frame buffer portion of the memory by the microprocessor or other client devices is provided through a virtual frame buffer device (VFBD) which recognizes the frame buffer portion of memory. Graphics operations, such as commands or data, generated by software operating on the microprocessor are intercepted by the VFBD which routes the commands or data through the memory controller to the frame buffer. The graphics controller also has access to the frame buffer portion of the shared memory. An arbitration mechanism, described in the above-referenced patent application, may be employed for arbitrating competing access requests to the shared memory from the memory controller and the graphics controller.

Brief Summary Text (20):
Also in the exemplary embodiment, an L1 cache is provided on the microprocessor and an L2 cache is provided connected to the memory controller. Cache coherency is maintained as follows. Data from the frame buffer portion of the shared memory is not cached in the L1 cache of the microprocessor. Otherwise, all accesses to the frame buffer by the graphics controller would require a snoop of the L1 cache. Moreover, if the L1 cache is a write back cache, the microprocessor must perform a flush operation after performing any accesses to the frame buffer memory. Likewise, if the L2 cache is a write back cache, the memory controller must perform a flush operation after completion of any access to the frame buffer.

Brief Summary Text (22):
In another alternative embodiment, the graphics controller is configured to perform only frame buffer refresh operations. All other graphics operations are performed by the microprocessor which routes the operations through the memory controller to the frame buffer portion of the shared memory.

Detailed Description Text (4):
FIG. 1 illustrates a computer system 10 having, as its principal components, a CPU or microprocessor 12, a shared memory 14, a memory controller 16 and a graphics controller 18. CPU 12 is connected to memory controller 16 by a system bus 39. The CPU, memory controller and graphics controller are interconnected via a PCI bus 24. One or more peripheral devices, such as bus master peripheral device 22, are also connected to the PCI bus. CPU 12 includes an on-board L1 cache 35. An external L2 cache 37 is connected to memory controller 16. A portion of memory 14 is dedicated as a frame buffer 20. Remaining portions of memory 14 are dedicated as system memory 21. The memory controller accesses both the system memory portion and the frame buffer portion of shared memory 14 based upon commands or data received from the CPU 12, peripheral device 22 or any other "client" device. The graphics controller accesses only the frame buffer portion of the shared memory. Both the memory controller and the graphics controller access the shared memory through a single memory bus 25.

Full	Title	Citation	Front	Review	Classification	Date	Reference			Claims	KWIC	Draw. D

☐ 13. Document ID: US 5854637 A

L1: Entry 13 of 15 File: USPT Dec 29, 1998

DOCUMENT-IDENTIFIER: US 5854637 A
TITLE: Method and apparatus for managing access to a computer system memory shared by a graphics controller and a memory controller

Brief Summary Text (10):
Hence, in such a system, both the memory controller and the graphics controller can access the same array of physical memory through a single interface bus. An arbitration mechanism must be employed to prevent conflicts between the memory controller and the graphics controller. An appropriate arbitration mechanism is set forth in the above-referenced copending patent application. A management system should also be employed to control access to the shared memory by software running on the microprocessor including operating system software and applications software. For example, operating system software must be prevented from corrupting the frame buffer portion of the shared memory while that portion of the shared memory is being used as a frame buffer by the graphics controller. Such corruption could occur if, for example, operating system software swaps data into or out of the frame buffer portion of the shared memory. On the other hand, to gain the benefit of having additional system memory when a frame buffer is not required, application software should be able to access all portions of the shared memory.

Brief Summary Text (14):
Another problem with implementing a shared computer system memory accessible through only a single interface bus is that access to the memory by the memory controller must be frequently blocked to allow the graphics controller to perform frame buffer refresh operations. In a typical proposed implementation, such frame buffer refresh operations must be performed fairly frequently. Accordingly, considerable bus bandwidth must be devoted to handling frame buffer refresh operations. Accordingly, it would be desirable to provide an improved method and apparatus for partitioning the system memory and the frame buffer memory within the shared memory to minimize the impact on total bus bandwidth by frame buffer refresh operations. Other aspects of the invention are drawn to such an improved partitioning.

Brief Summary Text (16):
A method and apparatus for managing access to a shared computer system memory accessible by both a memory controller and a graphics controller through a single interface bus is provided. In accordance with one aspect of the invention, the computer system is configured to selectively define a portion of the shared memory as a frame buffer. Graphics commands and data provided a microprocessor or other client device are routed through the memory controller if a frame buffer is created. Otherwise, graphics commands and data are routed through the graphics controller.

Brief Summary Text (17):
In one embodiment, the shared memory is partitioned between a system memory portion and a frame buffer memory portion with the frame buffer defined within a predetermined top-most portion of the physical memory. For example, in an eight megabyte memory system, the frame buffer may be defined as residing within the top one megabyte of the memory. At system start-up, a BIOS transmits a signal to an operating system of the microprocessor identifying the top of system memory as being the bottom of the frame buffer portion of memory. Accordingly, thereafter, operating system memory commands access only the system memory portion of the shared memory. Access to the frame buffer portion of the memory by the microprocessor or other client devices is provided through a virtual frame buffer device (VFBD) which recognizes the frame buffer portion of memory. Graphics operations, such as commands or data, generated by software operating on the microprocessor are intercepted by the VFBD which routes the commands or data through the memory controller to the frame buffer. The graphics controller also has access to the frame buffer portion of the shared memory. An arbitration mechanism,

described in the above-referenced patent application, may be employed for arbitrating competing access requests to the shared memory from the memory controller and the graphics controller.

Brief Summary Text (20):
Also in the exemplary embodiment, an L1 cache is provided on the microprocessor and an L2 cache is provided connected to the memory controller. Cache coherency is maintained as follows. Data from the frame buffer portion of the shared memory is not cached in the L1 cache of the microprocessor. Otherwise, all accesses to the frame buffer by the graphics controller would require a snoop of the L1 cache. Moreover, if the L1 cache is a write back cache, the microprocessor must perform a flush operation after performing any accesses to the frame buffer memory. Likewise, if the L2 cache is a write back cache, the memory controller must perform a flush operation after completion of any access to the frame buffer.

Brief Summary Text (22):
In another alternative embodiment, the graphics controller is configured to perform only frame buffer refresh operations. All other graphics operations are performed by the microprocessor which routes the operations through the memory controller to the frame buffer portion of the shared memory.

Detailed Description Text (4):
FIG. 1 illustrates a computer system 10 having, as its principal components, a CPU or microprocessor 12, a shared memory 14, a memory controller 16 and a graphics controller 18. CPU 12 is connected to memory controller 16 by a system bus 39. The CPU, memory controller and graphics controller are interconnected via a PCI bus 24. One or more peripheral devices, such as bus master peripheral device 22, are also connected to the PCI bus. CPU 12 includes an on-board L1 cache 35. An external L2 cache 37 is connected to memory controller 16. A portion of memory 14 is dedicated as a frame buffer 20. Remaining portions of memory 14 are dedicated as system memory 21. The memory controller accesses both the system memory portion and the frame buffer portion of shared memory 14 based upon commands or data received from the CPU 12, peripheral device 22 or any other "client" device. The graphics controller accesses only the frame buffer portion of the shared memory. Both the memory controller and the graphics controller access the shared memory through a single memory bus 25.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw D

14. Document ID: US 5818464 A

L1: Entry 14 of 15 File: USPT Oct 6, 1998

DOCUMENT-IDENTIFIER: US 5818464 A
TITLE: Method and apparatus for arbitrating access requests to a shared computer system memory by a graphics controller and memory controller

Detailed Description Text (3):
FIG. 1 illustrates a computer system 10 having, as its principal complements, a CPU 12, a shared physical memory 14, a memory controller 16 and a graphics controller 18. A portion memory 14 is dedicated as a frame buffer 20. Remaining portions of memory 14 are dedicated as system memory 21. The memory controller accesses memory 14 based upon commands received from the CPU 12 and from one or more peripheral devices, such as peripheral device 22 connected to the memory controller through a PCI bus 24. For example, the memory controller may read data from, and write data

to, memory 14. For some operations, such as a DRAM refresh, the memory controller requires access to all portions of memory 14. For other operations, the memory controller accesses only the system memory portion of shared memory 14. For still other operations, the memory controller accesses the frame buffer portion of shared memory 14. The above-referenced patent application provides details regarding a method and apparatus for managing access to the shared computer system memory through the memory controller and the graphics controller and such will not be described in detail herein.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. De

---

15. Document ID: US 5815167 A

L1: Entry 15 of 15 File: USPT Sep 29, 1998

DOCUMENT-IDENTIFIER: US 5815167 A
TITLE: Method and apparatus for providing concurrent access by a plurality of agents to a shared memory

Abstract Text (1):
A computer system, including a graphics controller and a memory controller, employs a Shared Frame Buffer Architecture, and accordingly has a shared memory in the form a bank of DRAMs. The shared memory is accessible by both the memory and graphics controllers. The memory includes a shared DRAM row in which a Shared Frame Buffer (SFB) aperture is defined. An interface selectively provides access to the shared DRAM row by the graphics or memory controller, while providing permanent access to the remaining DRAM rows by the memory controller. This facilitates concurrent access by the graphics controller and the memory controller to the shared DRAM row and to the remaining DRAM rows respectively, in a first memory access scenario. The accessibility of the shared DRAM row by the memory controller, in a second memory access scenario, is also maintained. The interface includes a selector circuit, such as a multiplexor or Q-switch, coupled to receive memory address signals and control signals from the graphics controller and the memory controller via a dedicated bus from each of these controllers. The selector circuit is operable selectively to present either memory address to the shared DRAM row, in which the SFB aperture is defined, and also selectively to provide access to the shared DRAM row by either controller. The selector circuit is operable by a logic circuit, incorporated within the systems controller, which determines whether a memory access request received from the memory controller is to an address in the shared DRAM row, or in the remaining DRAM rows.

Brief Summary Text (11):
According to a first aspect of the invention there is provided a computer system including a memory controller and a graphics controller. In one embodiment, the computer system employs a Shared Frame Buffer Architecture, and accordingly has a shared memory in the form of a bank of DRAMs. The shared memory is accessible by both the memory and graphics controllers. The shared memory includes at least one shared DRAM row in which a Shared Frame Buffer (SFB) aperture is defined. An interface selectively provides access to the SFB aperture by the graphics controller or the memory controller. This facilitates concurrent access to the SFB aperture by the graphics controller, and to the remaining DRAM rows by the memory controller while maintaining the accessibility to the at least one shared DRAM row by the memory controller. To reduce the likelihood of contention for access to the at least one shared DRAM row, the SFB aperture is preferably defined at a low memory location within the shared memory.

Brief Summary Text (13):
According to a second aspect of the invention there is provided a method of providing concurrent access by a graphics controller and a memory controller to a shared memory, the shared memory including a first memory portion and a second memory portion. The first memory portion, which may be single DRAM row, incorporates a Shared Frame Buffer (SFB) aperture. On receiving a request to access a first memory address in the first memory portion from the graphics controller, access to the first memory portion is granted to the graphics controller. Should a request to access a second memory address be received from the memory controller, a determination is made as to whether the second memory address is located in the first or second memory portions of the shared memory. Access to the second memory address is then granted to the memory controller if the second memory address is located in the second memory portion, so as to facilitate concurrent access by the graphics controller and the memory controller to the first and second memory portions respectively. On the other hand, if the second memory address is located in the first memory portion, access to the first memory portion by the memory controller is denied, or the contending requests for access to the first memory portion by the graphics and memory controllers are presented to an arbitration unit for arbitration.

Detailed Description Text (4):
The shared memory 20 is shown to comprise N rows of DRAM 20.1 to 20.N, each row of DRAM being coupled to receive control and address signals on lines 32.1 of the memory bus 32, and to receive and to output data on data lines 32.2 of the memory bus 32. Each row of DRAM can accordingly be accessed by either the memory controller 14 or the graphics controller 18, depending on which of these agents has control of the memory bus 32. DRAM row 20.N is shown to incorporate a shared frame buffer aperture 25, the frame buffer aperture 25 being designated for use by the graphics controller 18. The frame buffer aperture 25 is located at the top of the system memory 20 reported to the operating system, and memory above the frame buffer 25 may never be allocated by the operating system. It will be appreciated that the size and location of the frame buffer 25 within the shared memory 20 are definable and can be modified depending on the requirements of the computer system 10. Note that only a single set of control and address lines 32.1 and data lines 32.2 connect the shared memory 20 to the other components of the computer system 10 through a single port interface.

Detailed Description Text (9):
In the computer system 10 described above, access requests from the graphics controller 18 contend with memory access requests from the memory controller 14, which in turn may originate from a number of other components within the computer systems 10. It is furthermore important to note that the graphics controller 18 in fact only requires access to the single DRAM row 20.N incorporating the shared frame buffer 25. To acquire such access, the graphics controller 18 must have control of the single memory bus 32. Accordingly, when the graphics controller 18 has control of the memory bus 32, the memory controller 14 is precluded from issuing access requests originating within itself or received from any one of a number of other devices. Accordingly, system performance penalties may be suffered as a result of the lowered memory bandwidth resulting from the shared memory configuration.

Detailed Description Text (10):
Turning now to FIG. 2, there is shown a computer system 210 for implementing a method of providing concurrent access by at least two agents, such as a memory controller 214 and a graphics controller 218 to a shared memory 220. For the purposes of clarification, the shared memory 220 is shown to include only two memory portions, in the form of DRAM rows 220.1 and 220.2. A shared frame buffer aperture 225 is implemented in DRAM row 220.2. As with the computer system described above with reference to FIG. 1, the computer system 210 comprises a

processor 212 coupled to a host bus 222, which communicates with a peripheral bus 226 via a bus bridge 224. The bus bridge 224 incorporates a data path unit 228 and a system controller 230, in which the memory controller 214 and a memory arbitration unit 216 may be implemented. The arbitration unit 216 receives memory request signals (MREQ#) via line 235 from the graphics controller 218 and issues memory access grant signals (MGNT#) on line 236.

Detailed Description Text (13):
The interface 240 facilitates concurrent access by the memory controller 214 and the graphics controller 218 to the shared memory 220 when the memory controller 214 requests access to a memory location in a first portion of the shared memory 220, namely DRAM row 220.1, and when the graphics controller 218 requests access to a memory location in a second portion of the shared memory 220, namely DRAM row 220.2. Should the memory controller 214 require access to a memory location in DRAM 220.1, the switches 246.1 and 246.2 within the interface 240 are switched to a first state, by asserting an appropriate signal on line 248, this signal having been generated by the logic circuitry 231 incorporated within the system controller 230. Q-switch 246.1 provides a data path between data lines 244.2 and 232.2, so as to allow data to be propagated between the data path unit 228 and DRAM row 220.1. The memory controller 214 is able to address DRAM row 220.1 via control and address line 244.2. Similarly, Q-switch 246.2 creates a signal path between control and address line 234.1 and 242.1, so as to allow the graphics controller 218 to control and access DRAM row 220.2. The data can then be propagated between DRAM row 220.2 and the graphics controller 218 via data lines 234.2. In this way, when graphics controller 218 is accessing DRAM row 220.2, memory controller 214 is concurrently able to access DRAM row 220.1. Similarly, when memory controller 214 is accessing DRAM row 220.1, graphics controller 218 is concurrently able to access the shared buffer aperture 222 in DRAM row 220.2.

Detailed Description Text (15):
The present invention is not limited to a shared memory 220 incorporating only two DRAM rows. An alternative embodiment of the present invention is shown in FIG. 3. The shared memory 320 of a computer system 310 incorporates N DRAM rows, with the shared frame buffer aperture 325 being implemented in DRAM row 320.N. A row of DRAMs can be implemented using various combinations of DRAM banks. For example, a row of DRAM could compare a single 8 MB bank, or alternatively, two 4 MB banks. Applying the principals described above with reference to FIG. 2, it will be appreciated that an interface 340 is coupled to facilitate concurrent access by a graphics controller 318 to DRAM row 320.N and by a memory controller 314 to any of the other DRAM rows within the shared memory 320. The interface 340 accordingly views the shared memory 320 as being divided into two portions, namely DRAM row 320.N which incorporates the shared frame buffer aperture 322, and the remaining DRAM rows.

Detailed Description Text (16):
FIG. 4 shows a further alternative embodiment of the present invention, wherein the computer system 410 incorporates a 128-bit data path unit 428, which is coupled to receive and transmit data on the 64-bit data lines 442.2 and 444.2, thus obviating the need for two Q-switches within an interface unit 440. The data path unit 428 is coupled to receive a select signal 348 from system controller 430, thus allowing the data path unit 428 selectively to establish a data path between an external device and either the DRAM row 420.N incorporating the shared frame buffer aperture 422, or any of the other DRAM rows within the shared memory 420. The shared frame buffer 422 furthermore need not be located at the top of system memory, as illustrated in FIG. 4. As operating system loads and accesses are often to memory addresses located at or adjacent the top of the memory, it is desirable to locate the shared frame buffer 422 at a memory location just above the DOS application region within the memory 420. By locating the shared frame buffer 422 at a lower location within the memory 420, the probability of the memory controller 414 and the graphics controller 418 requesting access to the same DRAM row is lessened.

Accordingly, the concurrent access capability of the present invention can be more fully utilized.

Detailed Description Text (17):
The above described embodiments have focused on providing concurrent access to a shared memory incorporating a Shared Frame Buffer (SFB) by a graphics controller and a memory controller. It will be appreciated that other buffers could also be implemented within a shared memory, such as a so-called "Alpha Buffer" and a "Z Buffer", which contain information relating to the display of three-dimensional graphics. Accordingly, the teachings of the present invention can be extended to include providing concurrent access to a shared memory resource incorporating the above buffers. Furthermore, this concurrent access could be provided to other types of controllers, and to more than only two controllers or agents.

Detailed Description Text (18):
It will also be appreciated that the Shared Frame Buffer (SFB) need not be located within a single DRAM row as discussed above, but could be fragmented and located on a plurality of DRAM rows. In this case, the teachings of the present invention can be extended to provide concurrent access by a graphics controller to those DRAM rows in which the SFB resides, and by the memory controller to the other DRAM rows.

Detailed Description Text (20):
The present invention provides a number of advantages over prior art computer systems employing a shared memory. Most significantly, the present invention increases the effective memory bandwidth of a computer system with a shared memory, by allowing concurrent access to the shared memory by at least two agents, such as a graphics controller and a memory controller. The total system performance is thus comparable to that of a computer system having a dedicated memory resource for each of the requesting agents, whereas the costs benefits resulting from a shared memory are maintained. This advantage is achieved by increasing the granularity, or resolution, with which agents can access a shared memory resource. Thus, memory access contention between two agents is reduced to contention for individual memory rows, as opposed to contention for the entire shared memory. The ability of the present invention to provide a shared memory at a lower granularity is accomplished in part by providing separate control inputs to first and second portions of the shared memory. More specifically, the present invention proposes providing a dedicated memory bus for a row of DRAM incorporating a shared buffer aperture, and a dedicated memory bus for the rest of the rows of DRAM. The present invention also ensures a full 64-bit interface with the shared memory without increasing the pin count of a data path unit.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents
(unif$5 or shar$3) near3 (cache or buffer) same memory controller same graphics controller	15

Display Format: - Change Format

Previous Page Next Page Go to Doc#